LIF

Laboratoire d'Informatique Fondamentale
de Marseille

Unité Mixte de Recherche 6166
CNRS – Université de Provence – Université de la Méditerranée

On the partition of interval graphs into proper interval subgraphs

Frédéric Gardi

Rapport/Report 01-2002

Avril 2002

On the partition of interval graphs into proper interval subgraphs

Frédéric Gardi

LIF – Laboratoire d'Informatique Fondamentale de Marseille
UMR 6166
CNRS – Université de Provence – Université de la Méditerranée

Parc Scientifique et Technologique de Luminy,
163, avenue de Luminy - Case 901,
F-13288 Marseille Cedex 9, France.

gardi@lidil.univ-mrs.fr

Abstract/Résumé

In this paper the problem of the partition of an interval graph into proper interval subgraphs is considered. It arises in a study about the problem of working schedules planning. Here we give some upper bounds on the minimum size of such a partition as well as some efficient algorithms to compute it. In particular, we prove that every $K_{1,t}$-free interval graph, $t \geq 3$, is partitionnable into $\lfloor \frac{t}{2} \rfloor$ proper interval subgraphs. The proof of this proposition yields a linear or quasi-linear time algorithm to compute such a partition. Moreover, we give a linear-time algorithm to compute the minimum t such that an interval graph is $K_{1,t}$-free. Next, we show that each n-vertex interval graph can be partitionned into $O(\log n)$ proper interval subgraphs always in linear or quasi-linear time. Finally, we construct interval graphs for which this bound is sharp. **Keywords:** graph partition, working schedules planning, interval graphs, proper interval graphs, linear-time algorithms.

Nous abordons dans cet article le problème de la partition d'un graphe d'intervalles en sous-graphes d'intervalles propres. Celui-ci intervient dans une étude sur la problématique de la planification d'horaires de travail. Nous donnons ici des bornes sur la taille minimum d'une telle partition ainsi que des algorithmes efficaces pour la calculer. En particulier, nous prouvons que tout graphe d'intervalles sans $K_{1,t}$, $t \geq 3$, peut être partitionné en $\lfloor \frac{t}{2} \rfloor$ sous-graphes d'intervalles propres. Nous donnons un algorithme linéaire ou quasi-linéaire en temps pour déterminer une telle partition dans la preuve de cette proposition. De plus, nous proposons un algorithme linéaire en temps pour calculer le minimum t pour lequel un graphe d'intervalles est sans $K_{1,t}$. Ensuite, nous montrons que tout graphe d'intervalles à n sommets peut être partitionné en $O(\log n)$ graphes d'intervalles propres, toujours en un temps linéaire ou quasi-linéaire. Enfin, nous construisons des graphes d'intervalles pour lesquels cette borne est atteinte. **Mots-clés:** partition de graphes, planification d'horaires de travail, graphes d'intervalles, graphes d'intervalles propres, algorithmes linéaire en temps.

Relecteurs/Reviewers: Michel Van Canegehem, Victor Chepoi.

1 Introduction and preliminaries

An undirected graph $G = (V, E)$ is an *interval graph* iff to each vertex $v \in V$ an open (resp. closed) interval I_v in the real line can be associated, such that for each pair of vertices $u, v \in V$, $u \neq v$, $uv \in E$ if and only if $I_u \cap I_v \neq \emptyset$. Here we consider only *open* intervals but all our results can be easily extended to closed intervals. For an interval graph $G = (V, E)$, an *interval representation* of G will be noted $\{I_v\}_{v \in V}$ with $l_v, r_v \in \mathbb{R}$ the left and right endpoints of I_v and $|I_v|$ its size (i.e. its length). G is called a *proper interval graph* iff there is an interval representation for G in which no interval properly (strictly) contains another. In the same way, G is called an *unit interval graph* iff there is an interval representation for G in which each interval has unit size. Interval graphs arise in many practicle applications because they modelize many structures of the real-life world. They appear notably in areas like genetics, psychology, sociology, archæology, scheduling and others. The interested reader can consult [4] and [8] for surveys. In this paper, interval graphs are used to model a problem of working schedules planning, which can be defined as follows.

WSP problem:

Let $T_1, \ldots, T_n$ be n tasks such as $T_i = (l_i, r_i)$ where $l_i, r_i \in \mathbb{N}$ are the starting and ending dates of T_i. Let $m \in \mathbb{N}$ be the number of employees available and qualified to execute these tasks. Knowing that the tasks allocated to an employee must not overlap and the reglementation imposes no more than $k \in \mathbb{N}$ tasks by employees, are there enough employees to execute all the tasks ?

We will see that this problem can be formulated in graph-theoretic terms as a problem of coloring of an interval graph such that each colour is used at most k times. But before, we must remind some useful definitions from graph theory. Let $G = (V, E)$ be an undirected graph. We denote by *Adj*(v) the set of neighboors of a vertex $v \in V$ and by $d(v)$ the cardinality of *Adj*(v) (i.e. the *degree* of v). Given a subset $A \subseteq V$ of the vertices, we define the *subgraph induced by* A to be $G_A = (A, E_A)$, where $E_A = \{xy \in E \mid x \in A \text{ and } y \in A\}$. A *clique* of G is a set $C \subseteq V$ such that for all $x, y \in C$, $xy \in E$. A clique C is *maximal* if no clique of G properly contains C as a subset. A *stable set* (shortly a *stable*) of G is a set $S \subseteq V$ such that for all $x, y \in S$, $xy \notin E$. $\alpha(G)$ denotes the size of the largest stable in G. A *q-coloring* or a *partition of size q into stables of G*, $q \in \mathbb{N}$, is a partition $\mathcal{S} = \{S_1, \ldots, S_q\}$ of V such that each S_i is a stable. In the same way, we can define a *partition of size q into cliques of G* a partition $\mathcal{C} = \{C_1, \ldots, C_q\}$ of V such that each C_i is a clique. $\kappa(G)$ denotes the size of the partition of G into the least number of cliques. At last, a graph will be called *$K_{1,t}$-free* if it does not contain $K_{1,t}$ as an induced subgraph, $t \in \mathbb{N}$ (see Fig. 1).

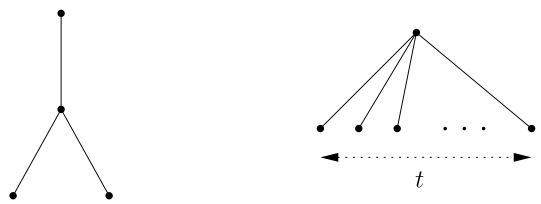

Fig. 1. The graphs $K_{1,3}$ and $K_{1,t}$.

F.S. Roberts has shown that the following classes of interval graphs coincide (see [7, 2, 3] for different proofs):

- proper interval graphs,
- unit interval graphs,
- $K_{1,3}$-free interval graphs.

To conclude, let us cite three fundamental properties of interval graphs which we will use later (see [4] for proofs).

Proposition 1 (Gilmore and Hoffman, 1964) *Let $G = (V, E)$ be an interval graph. There is a linear order $<$ on the maximal cliques of G such that for each vertex $v \in V$, the maximal cliques containing v occur consecutively in this order.*

Proposition 2 (Hajnal and Suràny, 1958) *If G is an interval graph, then $\alpha(G) = \kappa(G)$.*

Proposition 3 (Booth and Leuker, 1976) *Let G be an interval graph with n vertices and m edges. An interval representation of G such as the endpoints of intervals are in $\{0, \ldots, n\}$ can be computed in $O(n + m)$ time.*

Notice that a much simpler linear-time algorithm to compute such an interval representation of G is given in [6].

2 Motivations

Having reminded these definitions, we can clearly see that the WSP problem is equivalent to the problem of the partition of an interval graph by some stables of size at most $k \in \mathbb{N}$. Unfortunately, H.L. Bodlaender and K. Jansen have shown that this problem is $\mathcal{NP}$-complete even if k is a constant such that $k \geq 4$ [1]. On the other hand, in [3] it is proved that if the graph of intersections of the tasks is $K_{1,3}$-free (i.e. a proper interval graph) then the WSP problem can be solved in $O(n \log n)$ time by a greedy heuristic. Moreover, this one generates plannings which have some good properties *increasing their robustness facing the different risks of real-life situations* (delays of tasks, insertions/suppressions of tasks). So an idea is to partition the set of tasks initially hardly tractable into subsets which we will be able to treat better. That's why in this paper we are interested in the problem of the partition of an interval graph G into proper interval subgraphs. Especially, we give some bounds on the minimum size of this partition which we will denote by $p(G)$. First, we prove that every $K_{1,t}$-free interval graph, $t \geq 3$, is partitionnable into $\lfloor \frac{t}{2} \rfloor$ proper interval subgraphs. The proof of this proposition yields a linear or quasi-linear time algorithm to compute such a partition. Moreover, we give a linear-time algorithm to compute the minimum t such that an interval graph is $K_{1,t}$-free. Next, we show that each n-vertex interval graph can be partitionned into $O(\log n)$ proper interval subgraphs always in linear or quasi-linear time. Finally, we construct interval graphs for which this bound is sharp.

3 The main results

Here is a sufficient condition which gives us a first upper bound for $p(G)$.

Theorem 1 *Let $G = (V, E)$ be an interval graph with n vertices and m edges. If G is $K_{1,t}$-free with $t \geq 3$ then $p(G) \leq \lfloor \frac{t}{2} \rfloor$. Moreover, a partition of G into $\lfloor \frac{t}{2} \rfloor$ proper interval subgraphs can be done in $O(n + m)$ time (given G by adjacency lists in input) or $O(n \log n)$ time if we have an interval representation of G (i.e. given G by the list of endpoints of intervals in input).*

Proof. The following algorithm computes a partition of G into $\lfloor \frac{t}{2} \rfloor$ $K_{1,3}$-free interval subgraphs (i.e. proper interval subgraphs by Roberts' theorem) if G is $K_{1,t}$-free, $t \geq 3$.

Algorithm F
Input: a $K_{1,t}$-free interval graph G with $t \geq 3$;
Output: a partition of G into $\lfloor \frac{t}{2} \rfloor$ proper interval subgraphs;

```
begin;
    compute a minimum partition of G into cliques;
    let C_1, ..., C_κ(G) be such a partition linearly ordered according to <;
    let V_1, ..., V_⌊t/2⌋ be some sets of vertices;
    V_1 ← ∅, ..., V_⌊t/2⌋ ← ∅ and i ← 1;
    for j from 1 to κ(G) do
        V_i ← V_i ∪ C_j;
        i ← i + 1;
        if i > ⌊t/2⌋ do
            i ← 1;
    for i from 1 to ⌊t/2⌋ do
        let G_i be the subgraph of G induced by V_i;
    return G_1, ..., G_⌊t/2⌋;
end;
```

Algorithm F runs in linear or quasi-linear time since computing $C_1, \ldots, C_{\kappa(G)}$ can be done in $O(n + m)$ time [4] or $O(n \log n)$ time if we have an interval representation of G [5]. Now, we must prove its correctness, i.e. that each G_i returned by the algorithm is a $K_{1,3}$-free interval graph. Let $\{C_1^i, \ldots, C_k^i\}$, $k \geq 1$, be the set of cliques of G which have been assigned to V_i, linearly ordered according to $<$. We are going to prove that V_1 induces no copy of $K_{1,3}$ by an induction on the choice of C_j^1 made by Algorithm F. First, the clique C_1^1 included in V_1 cannot of course induce $K_{1,3}$. Now supposing that $\{C_1^1, \ldots, C_j^1\}$ induces a $K_{1,3}$-free interval graph, we can prove that $\{C_1^1, \ldots, C_{j+1}^1\}$ induces a $K_{1,3}$-free interval graph too for $j \in \{1, \ldots, k-1\}$ (see Fig. 2). Here we consider an interval representation of G. Let us suppose that there are four intervals $I, X \in C_{j-1}^1$, $Y \in C_j^1$ and $Z \in C_{j+1}^1$ inducing $K_{1,3}$ in $\{C_1^1, \ldots, C_{j+1}^1\}$. Reminding that the cliques of G are linearly ordered according to $<$, there are, by Algorithm F, $\lfloor \frac{t}{2} \rfloor$ cliques between C_{j-1}^1 and C_j^1 and still $\lfloor \frac{t}{2} \rfloor$ cliques between C_j^1 and C_{j+1}^1. Then $\alpha(G) = \kappa(G)$ (Proposition 2) implies that there is a stable of size $2\lfloor \frac{t}{2} \rfloor + 1$ with $t \geq 1$, which each interval is intersected by the interval I. Now $2\lfloor \frac{t}{2} \rfloor + 1 \geq t$, which implies the existence of the induced subgraph $K_{1,t}$ in G: a contradiction.

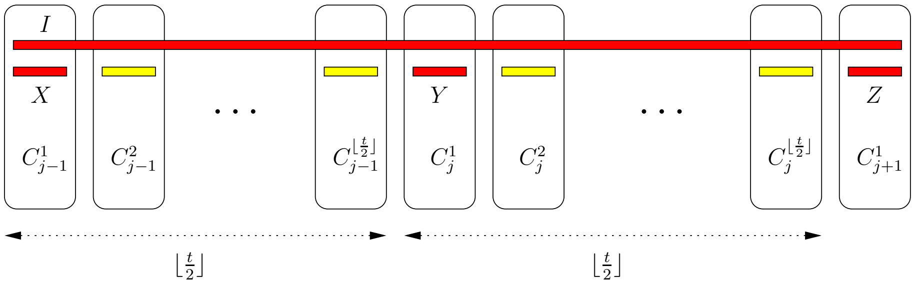

Fig. 2. An illustration of $K_{1,3} \Rightarrow K_{1,t}$ in the proof of Theorem 1.

So, we prove effectively that G_1 contains no induced copy of $K_{1,3}$, i.e. is a proper interval graph. By applying the same proof technic to $V_2, \ldots, V_{\lfloor \frac{t}{2} \rfloor}$, we obtain the whole correctness of Algorithm 1.

$\diamondsuit$

Corollary 1 *Let G be an interval graph with n vertices.*

$$p(G) \leq \lfloor \frac{n}{2} \rfloor$$

Proof. G is trivially $K_{1,n}$-free, so we can employ Theorem 1 with $t = n$.

$\diamondsuit$

Next, we will see that one can improve this last result for $p(G)$ by establishing a logarithmic upper bound. Before this, we will give an efficient algorithm to compute the minimum value t such that an interval graph G is $K_{1,t}$-free, in order to apply Theorem 1 at best for any interval graphs.

Algorithm G
Input: $G = (V, E)$ an interval graph;
Output: the minimum t such that G is $K_{1,t}$-free;

```
begin;
    compute an interval representation {I_v}_{v∈V} of G;
    t ← 0;
    for each v ∈ V do
        order Adj(v) according to the increasing right endpoints;
        compute α(G_v) where G_v is the subgraph induced by Adj(v);
        if t < α(G_v) do
            t ← α(G_v);
    return t+1;
end;
```

Theorem 2 *Let $G = (V, E)$ be an interval graph with n vertices and m edges. Algorithm G determines the minimum t such that G is $K_{1,t}$-free in $O(n + m)$ time or $O(n^2)$ time if we have an interval representation of G.*

Proof. The correctness of the algorithm comes from the definition of a $K_{1,t}$-free graph. By Proposition 3, computing an interval representation of G such that each endpoint of the intervals are in $\{0,\ldots,n\}$ can be done in $O(n+m)$ time [4, 6]. Now, in the loop, ordering $Adj(v)$ can be done in $O(d(v))$ time by using $O(d(v))$ space. Thus, a maximum stable of G_v can be computed in $O(d(v))$ time [5]. Consequently, the total time complexity of Algorithm G is in $O(n+m)$. In the same way, we can prove that if we have an interval representation of G, the algorithm runs in $O(n^2)$ time.

$\diamondsuit$

Now, we will prove the promised logarithmic upper bound for $p(G)$.

Theorem 3 *Let G be an interval graph with n vertices.*

$$p(G) \leq \lfloor \tfrac{t+2}{2} \rfloor \lceil \log_t n \rceil \text{ with } t \in \{2,\ldots,n-2\}.$$

Proof. Let $\mathcal{I} = \{I_1,\ldots,I_n\}$ be an interval representation of G such that $\forall i$, $|I_i| \in \mathbb{N}$. Let $l = \max_{I_i \in \mathcal{I}} |I_i|$. We claim that $p(G) \leq \lfloor \frac{t+2}{2} \rfloor \lceil \log_t l \rceil$ with $t \in \{2,\ldots,n-2\}$. Indeed, we can partition the set of intervals $\mathcal{I}$ into $\lceil \log_t l \rceil$ subsets $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_{\lceil \log_t l \rceil}$ such that:

- $\mathcal{I}_1$ contains the intervals I_i of $\mathcal{I}$ of size $|I_i| \in \{1,\ldots,t\}$,
- $\mathcal{I}_2$ contains the intervals I_i of $\mathcal{I}$ of size $|I_i| \in \{t,\ldots,t^2\}$,

  ⋮
- $\mathcal{I}_{\lceil \log_t l \rceil}$ contains the intervals I_i of $\mathcal{I}$ of size $|I_i| \in \{t^{\lceil \log_t l \rceil - 1},\ldots,l\}$.

We affirm that each set $\mathcal{I}_j$, $j \in \{1,\ldots,\lceil \log_t l \rceil\}$, does not contain the induced subgraph $K_{1,t+2}$. Indeed, for each $\mathcal{I}_j$, we have $\max_{I_i \in \mathcal{I}_j} |I_i| \leq t \min_{I_i \in \mathcal{I}_j} |I_i|$, which implies that no interval of $\mathcal{I}_j$ contains properly a stable of size t. Now let us suppose that a set of intervals of $\mathcal{I}_j$ induces $K_{1,t+2}$, this would imply that a stable of size at least t is properly contained in one interval of $\mathcal{I}_j$, in contradiction with the previous affirmation. Consequently, by Theorem 1, we can partition the graph induced by each subset $\mathcal{I}_j$ into $\lfloor \frac{t+2}{2} \rfloor$ proper interval subgraphs, which proves the claim. Now, by Proposition 3, there is an interval representation of G such that the endpoints of each interval are some integers and $l = n$, which allows us to conclude.

$\diamondsuit$

Corollary 2 *Let G be an interval graph with n vertices.*

$$p(G) \leq 2(\log_3 n + 1).$$

Moreover, a partition of G into $2(\log_3 n+1)$ proper interval subgraphs can be obtained in $O(n+m)$ time or $O(n \log n)$ time if we have an interval representation of G.

Proof. By Theorem 3, we have $p(G) \leq \lfloor \frac{t+2}{2} \rfloor (\log_t n + 1)$ with $t \in \{2, \ldots, n-2\}$. Now, the minimum of the function $\frac{1}{\ln t} \lfloor \frac{t+2}{2} \rfloor$ is reached for $t = 3$ (see Fig. 3 below). Consequently, the best bound that we can obtain from Theorem 3 is $p(G) \leq 2(\log_3 n + 1)$. Moreover, the construction of the sets $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_{\lceil \log_3 l \rceil}$ can be done in $O(n)$ time by sorting the intervals according to their size. Indeed, by Proposition 3, we can compute in $O(n+m)$ time an interval representation of G such that the size of the intervals are in $\{0, \ldots, n\}$ [4, 6]. Thus sorting can be done in $O(n)$ time by using a space in $O(n)$. If we have an interval representation of G, the construction of these sets can clearly be done in $O(n \log n)$ time. Then, by Theorem 1, we can conclude as for the time complexity.

♢

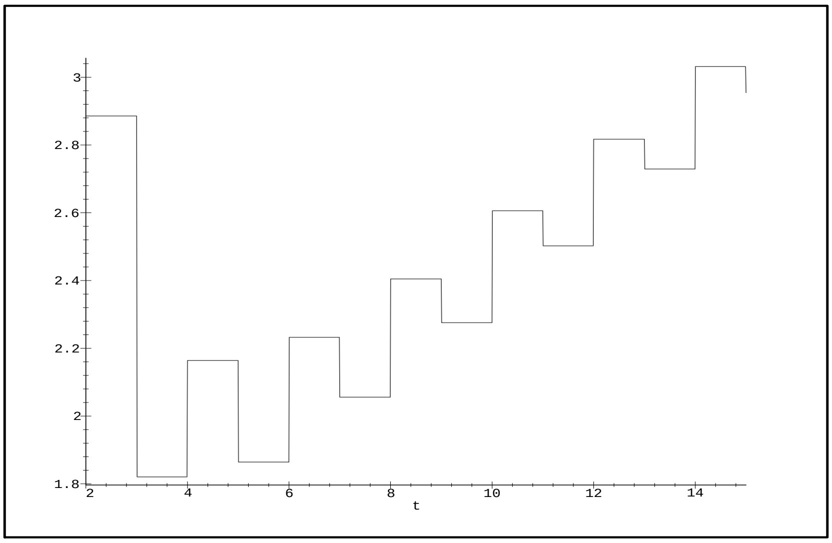

Fig. 3. The evolution of $\frac{1}{\ln t} \lfloor \frac{t+2}{2} \rfloor$ in accordance with t.

Finally, we will prove that this last bound is asymptotically almost reached.

Theorem 4 *There exist some n-vertex interval graphs G such that $p(G) = \log_3(2n+1)$.*

Proof. Here we will give the interval representation of an interval graph G_q with n_q vertices such that $p(G_q) = \log_3(2n_q + 1)$, $q \in \mathbb{N}$. This graph G_q owns q stables $S_1, \ldots, S_q$ constructed recursively. S_1 just consists of one interval of length $3^{(q-1)}$. Then, each S_i will be built like this: "clone" the stable S_{i-1} and subdivide each interval of this "cloned" stable into three intervals of equal size in order to include them in S_i (see Fig. 4 for an example).

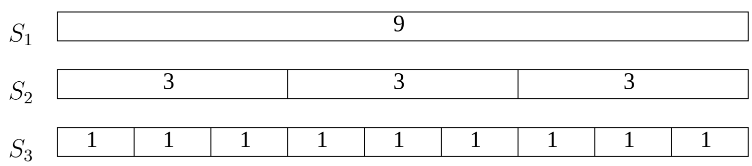

Fig. 4. An example of construction of G_q with $q = 3$.

Consequently, the number of intervals in G_q will be $n_q = \sum_{i=1}^{q} 3^{i-1} = \frac{3^q-1}{2}$ (i). Trivially, we can see that $p(G_q) \leq q$ (each stable of G_q induces a proper interval subgraph). Now we can show that G_q must be partitionned into at least q proper interval subgraphs. We can do an induction on $i \in \{1, \ldots, q\}$, the number of stables of G_i. Trivially $p(G_1) = 1$. Now supposing that $p(G_i) = i$ (ii), we are going to prove that $p(G_{i+1}) = i+1$ for $i \in \{1, \ldots, q-1\}$. For this, let us suppose on the contrary that $p(G_{i+1}) = i$: G is partitionnable into $\mathcal{I}_1, \ldots, \mathcal{I}_i$ sets of intervals which induces each of them a $K_{1,3}$-free interval graph. Without loss of generality, let us suppose that the unique interval $I \in S_1$ belongs to $\mathcal{I}_1$. In this case, the other intervals of $\mathcal{I}_1$ must make at most two cliques C_a and C_b with I because a third clique C_c implies clearly the existence of an induced copy of $K_{1,3}$ in $\mathcal{I}_1$ (see Fig. 5).

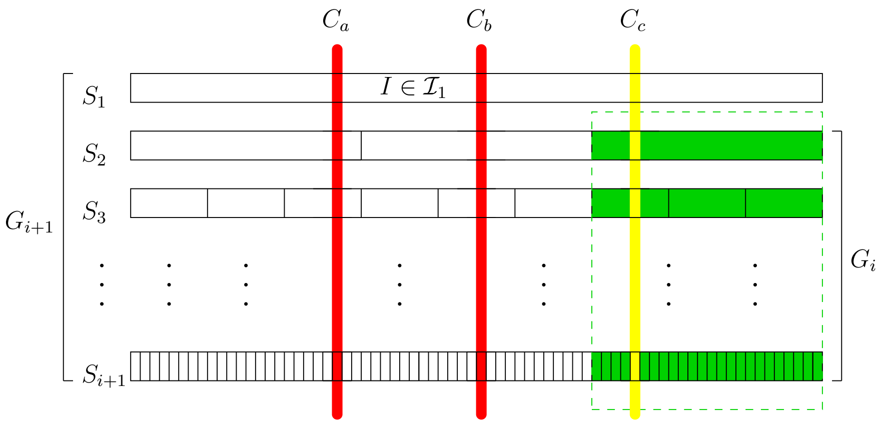

Fig. 5. An illustration of the proof of $p(G_{i+1}) = i + 1$.

Then there are necessarily an interval of S_2 and all the intervals of $S_3, \ldots, S_{i+1}$ stemming from its subdivision (the framed and colored intervals on Fig. 5) which don't belong to $\mathcal{I}_1$. This set of intervals induces clearly a graph G_i and by hypothesis (ii), needs i sets to be partitionned into proper interval subgraphs. Now, we have just $\mathcal{I}_2, \ldots, \mathcal{I}_i$, that is to say $i - 1$ sets, to realize this partition, which implies a contradiction. Consequently $p(G_{i+1}) = i+1$ for $i \in \{1, \ldots, q-1\}$ and so $p(G_q) = q$. Now by (i), we have $q = \log_3(2n_q + 1)$ which allows us to conclude.

$\diamondsuit$

Corollary 3 *There exist some $K_{1,t}$-free interval graphs G with n vertices such that $p(G) = \lfloor \log_3(t-1) \rfloor + 1$ for $t \in \{2, \ldots, n-1\}$.*

Proof. The graph G_q constructed in Theorem 4 is without $K_{1,3^{(q-1)}+1}$. Now, it suffices to state $t = 3^{q-1} + 1$ and we have $p(G_q) = q = \lfloor \log_3(t-1) \rfloor + 1$.

$\diamondsuit$

4 Conclusion

Table 1 relates some results about $p(G)$ for $K_{1,t}$-free interval graphs, obtain from Theorem 1, with some small values for t.

G has no induced copy of	theoretical upper bound for $p(G)$	reached bound for $p(G)$
$K_{1,3}$	1	1
$K_{1,4}$	2	2
$K_{1,5}$	2	2
$K_{1,6}$	3	2
$K_{1,7}$	3	2
$K_{1,8}$	4	2
$K_{1,9}$	4	2
$K_{1,10}$	5	3
$K_{1,11}$	5	3
$\vdots$	$\vdots$	$\vdots$
$K_{1,t}$-free	$\lfloor \frac{t}{2} \rfloor$	$\lfloor \log_3(t-1) \rfloor + 1$

Table 1. Some bounds that we obtain for $p(G)$.

Now we can give a general result about the size $p(G)$ of the minimum partition of an interval graph G into proper interval subgraphs.

Theorem 5 *Let G be an interval graph with n vertices.*

$$p(G) = O(\log n)$$

Moreover, this bound is asymptotically reached for some interval graphs and an $O(\log n)$-partition of G into proper interval subgraphs can be computed in linear or quasi-linear time.

We can note that this result gives immediately an $O(\log n)$-approximation for the following optimization problem.

Problem: *Partition of an interval graph into proper interval subgraphs.*
Input: *An interval graph G.*
Goal: *Find a partition of G into a minimum number of proper interval subgraphs.*

We conjecture that this problem is $\mathcal{NP}$-complete.

At last, we must precise that Theorem 5 can be easily extended to the class of circular-arc graphs which englobes the class of interval graphs (see [4] for an introduction and different applications of circular-arc graphs).

Acknowledgements

We would like to thank Professor Michel Van Caneghem and Professor Victor Chepoi for their precious advice and the firm *PROLOGIA - Groupe Air Liquide* for its grant.

References

[1] H.L. Bodlaender and K. Jansen. Restrictions of graph partition problems. Part I. *Theoretical Computer Science*, 148:93–109, 1995.

[2] K.P. Bogart and D.B. West. A short proof that "Proper = Unit". *Discrete Mathematics*, 201:21–23, 1999.

[3] F. Gardi. The mutual exclusion scheduling problem for proper interval graphs. (in preparation).

[4] M.C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Computer Science and Applied Mathematics. Academic Press, New-York, 1980.

[5] U.I. Gupta, D.T. Lee, and J.Y.-T. Leung. Efficient algorithms for interval graphs and circular-arc graphs. *Networks*, 12:459–467, 1982.

[6] M. Habib, R. McConnel, C. Paul, and L. Viennot. Lex-BSF and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing. *Theoretical Computer Science*, 234:59–84, 2000.

[7] F.S. Roberts. Indifference graphs. In Frank Harary, editor, *Proof Techniques in Graph Theory*, pages 139–146. Academic Press, New-York, 1969.

[8] F.S. Roberts. *Graph Theory and its Applications to Problems of Society.* SIAM, Philadelphia, 1978.